Prediction of Gas Concentration Using Gated Recurrent Neural Networks

Wang, Shu ; Hu, Yuhuang ; Burgues, Javier ; Marco, Santiago ; Liu, Shih-Chii

Abstract: Low-cost gas sensors allow for large-scale spatial monitoring of air quality in the environment. However they require calibration before deployment. Methods such as multivariate regression techniques have been applied towards sensor calibration. In this work, we propose instead, the use of deep learning methods, particularly, recurrent neural networks for predicting the gas concentrations based on the outputs of these sensors. This paper presents a first study of using Gated Recurrent Unit (GRU) neural network models for gas concentration prediction. The GRU networks achieve on average, a 44.69% and a 25.17% RMSE improvement in concentration prediction on a gas dataset when compared with Support Vector Regression (SVR) and Multilayer Perceptron (MLP) models respectively. With the current advances in deep network hardware accelerators, these networks can be combined with the sensors for a compact embedded system suitable for edge applications.

DOI: https://doi.org/10.1109/aicas48895.2020.9073806

Prediction of Gas Concentration Using Gated Recurrent Neural Networks

Shu Wang*, Yuhuang Hu*, Javier Burgués†‡, Santiago Marco†‡ and Shih-Chii Liu*
*Institute of Neuroinformatics, University of Zürich and ETH Zürich, Zurich, Switzerland
†Institute for Bioengineering of Catalonia, The Barcelona Institute of Science and Technology, Barcelona, Spain
‡Department of Electronics and Biomedical Engineering, Universitat de Barcelona, Barcelona, Spain
{shu,shih}@ini.uzh.ch

***Abstract*—Low-cost gas sensors allow for large-scale spatial monitoring of air quality in the environment. However they require calibration before deployment. Methods such as multivariate regression techniques have been applied towards sensor calibration. In this work, we propose instead, the use of deep learning methods, particularly, recurrent neural networks for predicting the gas concentrations based on the outputs of these sensors. This paper presents a first study of using Gated Recurrent Unit (GRU) neural network models for gas concentration prediction. The GRU networks achieve on average, a 44.69% and a 25.17% RMSE improvement in concentration prediction on a gas dataset when compared with Support Vector Regression (SVR) and Multilayer Perceptron (MLP) models respectively. With the current advances in deep network hardware accelerators, these networks can be combined with the sensors for a compact embedded system suitable for edge applications.**

I. Introduction

Deep networks are current machine learning algorithms that are achieving state-of-the-art results on many machine learning tasks. Most of the reported progress is in the area of vision and audio applications [1], [2]. These network architectures particularly convolutional and recurrent neural networks have been implemented on embedded platforms [3]–[9], thereby suitable for processing the outputs of sensors for edge applications.

The use of the current state-of-the-art deep networks together with sensors such as gas sensors is less explored. Gas sensors are important for environmental monitoring, in particular, to detect undesirable gases such as carbon monoxide even at low concentrations [10]–[12]. Two prevailing technologies for monitoring gases include metal oxide semiconductor (MOX) and electrochemical sensors. MOX sensors have a shorter response time and longer life time than electrochemical sensors. On the other hand, they consume relatively higher power and have less sensitivity.

Overall, because these sensors are low-cost and lightweight, they are ideal for pervasive low-power mobile sensing systems. However, these sensors have some disadvantages: They suffer from drift and are sensitive to the presence of other gases and environmental factors such as temperature and humidity [10], [13]–[15]. They also need to be calibrated before they are deployed in the field. Calibration methods include linear and nonlinear estimation methods [13], [16]. Nonlinear methods that are currently used include Support Vector Regression (SVR) [17], [18] and Multilayer Perceptron (MLP) networks [12], [19]. These models consistently perform better than linear methods [11], [16], [20].

This work was partially funded by the SNSF grant CRSII5_177255 and the National Centre of Competence in Research (NCCR) Robotics.

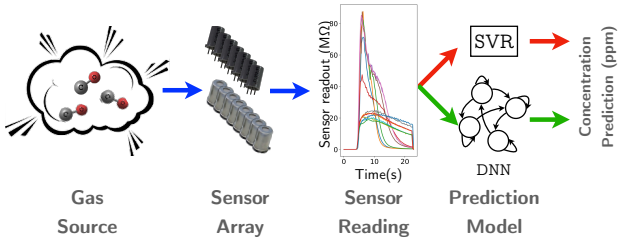

Fig. 1. Overview of the gas concentration prediction pipeline.

In this study, we look at the performance of a different deep learning architecture on gas concentration prediction using an available dataset recording from MOX sensors [15], [21]. Because the output readings of these sensors form a time series, a Gated Recurrent Neural Network (GRNN) is used for model calibration.

We investigate the advantage of GRNNs over other nonlinear algorithms (SVR, MLP) and the feasibility of using these networks for predicting gas concentrations in real-time. Section II presents the dataset and the different methods including the GRNNs used in this study. Section III presents the experiments and the results; and Section IV presents conclusions on the work.

II. Methods

Fig. 1 shows the pipeline of the processing described in this work. An array of gas sensors measures the concentration of a particular gas in a test chamber. The readings from the sensor array are then passed to a deep neural network (DNN) or a statistical model for prediction of the gas concentration.

We first present the three different models that will be used for predicting the gas concentration in a gas dataset described in Section III.

A. Support Vector Regression

The Support Vector Regression (SVR) method is the regression form of the Support Vector Machine (SVM) [22].

We used the off-the-shelf SVR implementation with the radial basis function (RBF) kernel in `ThunderSVM` library [23]. During training, we used a five-fold cross-validation for tuning the error tolerance term (the regularization parameter $C \in \{2^i\}_{i=-1}^{8}$) for SVR models.

B. Gated Recurrent Neural Network

In this work, we use a GRNN model with *Gated Recurrent Units* (GRUs) [24], [25]. The update equations of the GRU are as follows:

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z\right) \quad (1)$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r\right) \quad (2)$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}\right) \quad (3)$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t \quad (4)$$

where $\sigma(\cdot)$ is the sigmoid activation function, $\tanh$ represents the hyper-tangent activation function, and $\odot$ is an element-wise multiplication. Equations 1–4 represent the update gate, the reset gate, the candidate hidden state, and the hidden state respectively. $\{\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}\}$ are input-to-hidden weight matrices. $\{\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}\}$ are hidden-to-hidden matrices. $\{\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}\}$ are bias vectors. The output of the reset gate $\mathbf{r}_t$ is used to compute the candidate hidden state $\tilde{\mathbf{h}}_t$ and the output of the update gate $\mathbf{z}_t$ is used to update the hidden state $\mathbf{h}_t$.

Our regression model consists of a single GRU layer. It takes as input a sequence of sensor readings of length 83. The last hidden state is passed through a linear regressor to predict the gas concentration. Different GRU models with layer sizes of $\{50, 100, 200\}$ units are tested in this work.

C. Multilayer Perceptron

A Multilayer Perceptron (MLP) network is a feedforward network that consists of an input layer, a number of hidden layers and an output layer. The units in each layer are fully connected to the units in the next layer. MLP networks have been studied for compensating the sensor drift in ion-sensitive field-effect transistor (ISFET)-based sensors [26]. In this paper, we use a MLP network with two hidden layers that has the same number of hidden units. To match the similar number of parameters as in the GRU models, we tested two configurations where each hidden layer has either 15 or 170 units. The number of parameters for each model is presented in Table I.

D. Training Details and Evaluation Metrics

The MLP and GRU models are trained using Adam [27] with batch size 64. The learning rate is set to 10^{-4}. All models are implemented in `PyTorch` [28] and trained over 100 epochs. Each model is trained to minimize the mean-square error (MSE) between the prediction and actual concentration:

$$\mathrm{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (5)$$

where N is the number of samples, $\hat{y}_i$ is the i-th ground truth concentration, and y_i is the predicted concentration.

For all three types of models, the performance of the prediction is measured using the root-mean-square error (RMSE) of the prediction following:

$$\mathrm{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\mathrm{MSE}(\mathbf{y}, \hat{\mathbf{y}})} \quad (6)$$

The unit of the RMSE is parts per million (ppm) of concentration.

In addition, we report the coefficient of determination (R^2 score) of the best model on the test dataset.

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\mathrm{MSE}(\mathbf{y}, \hat{\mathbf{y}})}{\mathrm{MSE}(\bar{y}, \hat{\mathbf{y}})} \quad (7)$$

where $\bar{y} = \frac{1}{N}\sum_{i=1}^{N}\hat{y}_i$ is the mean of the ground truth data. R^2 reflects the quality of the regression model. This metric is particularly useful when RMSE does not reflect the regression performance (*e.g.*, predicting near-zero value may yield low RMSE yet bad R^2 score). As R^2 approaches 1, the quality of the regression is better.

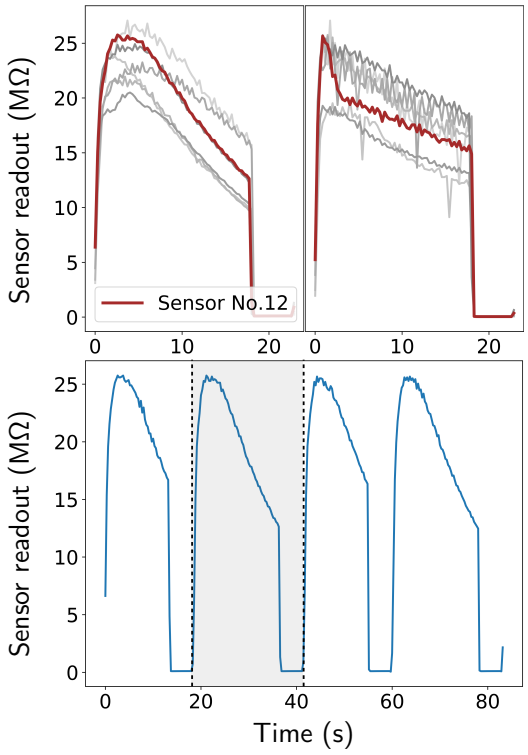

Fig. 2. Example of a segment of sensor recordings from the dataset. The top figures show two single sequences of recordings from the 7 FIS SB-500-12 sensors (depicted in different grays) at concentration of 17.76 ppm (left) and 2.22 ppm (right) respectively. The bottom figure shows the measurements during a segment of four repeated heating and cooling periods of Sensor No. 12. The shaded area corresponds to the time duration for the curves in the top left figure.

III. Experiments

A. Gas Sensor Dataset

We used a gas sensor dataset [15], [21] that contains recordings from a chemical detection platform which is composed of 14 temperature-modulated MOX gas sensors of two different

types (FIS SB-500-12 [29] and Figaro TGS 3870-A04 [30]). The sensors were operated by using a built-in heater which repeatedly heats and cools down the sensor. The resistance of the sensor was measured during this repeated period of heating and cooling (see Fig. 2 for an example). The sensor readings during the cooling down period were used to determine the carbon monoxide gas concentration during sensor operation.

The platform was exposed to dynamic mixtures of the target gas, carbon monoxide (CO), and humid synthetic air in a test chamber. The CO gas was presented to the sensors at 10 concentration levels (evenly distributed between 0 and 20 ppm inclusive). Temperature and reference humidity values inside the test chamber were also recorded with a temperature/humidity sensor.

The data acquisition took 13 measuring days and spanned over 17 days. The measurement protocol was the same for each measuring day. The daily measurements included 10 repeated measurements of each CO concentration. In each repetition, the relative humidity (r.h.) level was chosen randomly from a uniform distribution between 15% to 70%. In total 100 CO concentration-r.h combinations were performed in random order where every combination lasted approximately 15 minutes. Further details of this dataset acquisition can be found in [15], [21].

We used only the data recorded with the FIS SB-500-12 sensors for this work because the sensors were shown to be less sensitive to humidity changes compared to Figaro TGS 3870-A04 sensors [21]. In addition, we used the readings from all 7 sensors for a larger training set.

B. Dataset Preparation

The original long time series of the recordings in the gas sensor dataset were segmented into individual sample sequences each corresponding to one of the repeated heating and cooling cycles in the dataset. Every sample consists of readings starting from the transition from high to low of a heating cycle (see gray segment of curve in Fig. 2). The heating cycle was the concatenation of cycles of 20 s and 25 s, therefore, the sensor output patterns shall last either 20s (odd cycles) or 25 s (even cycles). The samples from odd cycles are zero-padded to match the length of the even cycle samples.

This processed dataset is then divided into a training set and a test set dependent on the three experiments carried out in this paper:

a) 70/30 Split: 70% of the sequences are randomly selected as part of the training set, while the rest are assigned to the test set.

b) 9/4 day Split: This split method keeps the time dependency of the samples. Sequences belonging to the first 9 measuring days are used for training, and the sequences from the remaining 4 days are used for testing.

c) 1st day/12 day Split: In this case, the model is trained on recordings from the first measuring day, and then tested on successive days.

GRU models receive each sequence along the temporal axis. At each time stamp, a 7-dimensional vector is processed by the GRU layer. Each sequence has 83 time stamps. For SVR and MLP models, each sequence is first flattened into a vector before fed into the model. Therefore, the input data dimension is $7 \times 83 = 581$.

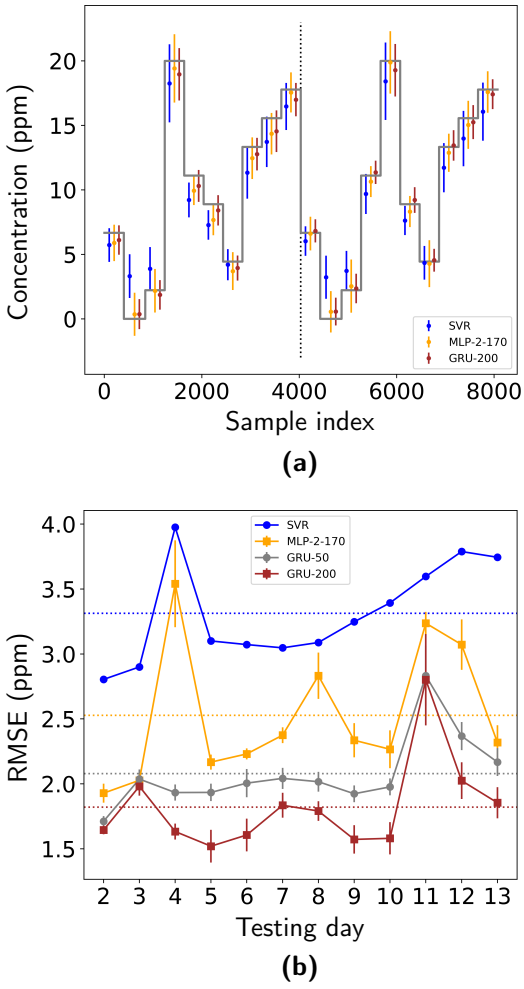

Fig. 3. **(a)** Predictions of the three methods for a recording sequence over the 12th and 13th measuring days. The vertical line shows the separation between recordings from separate days. Solid black line shows the actual concentrations presented to the sensors. **(b)** Predicted RMSEs for algorithms that are trained only on the first day and tested on subsequent days. The dotted lines represent the average RMSEs for respective models.

C. Model Prediction Results

Table I summarizes the prediction results of the three methods. Overall, the GRU models perform better than the SVR and MLP models. In terms of RMSE, GRU models outperform SVR and MLP models by 45.15% and 25.17% on average. The R^2 scores of GRU models are also higher than SVR and MLP models by an average of 15.86% and 5.19% respectively. Increasing the size of the GRU layer also leads to a decreasing RMSE prediction. As expected, GRU models trained on the 70/30 Split dataset report the best prediction results because the networks receive information from all 13 measuring days during training. Surprisingly, the worst-performing GRU model (GRU-50 trained with 1st day/12 day

TABLE I

PREDICTION RESULTS OVER 10 TRIALS. REPORTED ARE THE MEAN AND STANDARD DEVIATION OF RMSE (PPM) AND THE COEFFICIENT OF DETERMINATION R^2 FOR EACH TRIAL. STANDARD DEVIATION VALUES FOR SVR MODELS ARE OMITTED BECAUSE THEY ARRIVE AT THE SAME SOLUTION FOR THE SAME HYPERPARAMETER VALUE C DURING CROSS VALIDATION.

Model	No. Parameters	70/30 Split		9/4 day Split		1st day/12 day Split	
		RMSE	R^2	**RMSE**	R^2	**RMSE**	R^2
SVR	N/A	2.192	0.884	2.593	0.837	3.333	0.731
MLP-2-15	9.0 K	1.753±0.028	0.926±0.002	2.014±0.037	0.902±0.004	2.937±0.130	0.791±0.020
MLP-2-170	128.2 K	1.531±0.013	0.943±0.001	1.934±0.018	0.910±0.002	2.577±0.109	0.839±0.013
GRU-50	8.9 K	1.334±0.026	0.957±0.002	1.775±0.109	0.923±0.010	2.098±0.069	0.893±0.007
GRU-100	32.8 K	1.225±0.027	0.964±0.002	1.642±0.052	0.935±0.004	1.923±0.049	0.910±0.005
GRU-200	125.6 K	1.123±0.020	0.969±0.001	1.530±0.034	0.943±0.003	1.856±0.061	0.916±0.006

Split) still shows considerably lower RMSE and higher R^2 score than the best-performing SVR model trained on the 70/30 Split dataset.

Fig. 3(a) shows the model comparison among the best performing SVR, MLP and GRU models using the 9/4 day Split dataset. The concentration predictions of the GRU model (red symbols) is visually better than predictions of SVR (blue symbols) and MLP models (yellow symbols). The SVR model seems to over-predict when the gas concentration is low. In Fig. 3(b), we plot the predicted RMSEs of individual days using the 1st day/12 day Split dataset. The SVR and MLP models yield higher RMSE values and the RMSE for SVR shows a trend of increasing prediction error as the measuring day progresses. This trend of increasing prediction error might be caused by the known drift problem of these sensors. On the other hand, the GRU model does not show this trend and could be less invariant to the sensor drift in this dataset. Note that the GRU model does not perform well on the 11th day's recording. After investigation, we see that the changing humidity patterns on this day are different than from other days. Similarly, the SVR and MLP models behave worse on 4th day's recording. This is because the temperature pattern on the 4th day's recording also differs from other days. It shows that these models are prone to the variability in humidity and temperature.

D. Hardware Performance

Table II shows the runtime for different size models on a GPU which dissipates 7.5 W and two RNN hardware accelerators [7], [31]. While the runtime of the GRU models are benchmarked on the GPU, the runtime of the models on the DeepStore [7] and EdgeDRNN [31] accelerators are estimated from the reported throughput numbers.

Unlike SVM and MLP models that produce the prediction value only after getting the full sequence, GRU models can distribute the computation over time because the hidden state is updated for each input sample that arrives. The nature of the temporal computation in an RNN reduces the memory needed to store all samples of the sequence. Table II shows that the inference time per recurrent update grows with the model size for the FPGA accelerators. Running a small model on an embedded platform is preferred but a smaller model can lead to lower accuracy for regression as shown in Table I. If this accuracy is acceptable as in the case of the GRU-50 model, results show that running this model on the EdgeDRNN is 64x faster than using a Jetson TX2 GPU.

TABLE II

HARDWARE INFERENCE TIME PER RECURRENT UPDATE (μs). SHOWN ARE ESTIMATED VALUES USING EDGEDRNN (4.1 GOPS/S) [31] AND DEEPSTORE (1.04 GOPS/S) [7]; AND MEASURED VALUES ON A JETSON TX2 GPU.

	EdgeDRNN	**DeepStore**	**Jetson TX2**
GRU-50	4.22	16.56	271.99
GRU-100	15.76	61.83	271.15
GRU-200	60.78	238.51	271.36

IV. CONCLUSION

In this paper, we present the first study of gas concentration prediction using deep recurrent neural networks. The network is trained on a gas dataset which contains measurements from MOX gas sensors in the presence of different CO concentrations. The results show that GRU models on average, show a 44.69% and 25.17% improvement over SVR and MLP models respectively as measured by the RMSE of the prediction. GRU models have an advantage over SVR and MLP models because every incoming sample can be used to update GRU models whereas the whole sequence needs to be stored for the other two. The memory storage requirements would therefore increase as the length of the input temporal sequence increases.

In this work, we directly used the raw sensor's readings. Future work would look at the application of the preprocessing of the sensors reading as proposed in [14], [15], [21]. The cost of adding this preprocessing step for an online system will also be considered.

Because we intend that the model be deployed on an edge device, the runtime of the GRU-based model will also improve based on the recent developments in RNN hardware accelerators [7], [9], [31]. By implementing the network on this hardware platform, we can aim for a smart embedded system for online air quality monitoring in the environment.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[2] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[3] L. Cavigelli and L. Benini, "Origami: A 803-GOp/s/W convolutional network accelerator," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2461–2475, Nov 2017.

[4] Y. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan 2017.

[5] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, and Y. Chen, "Cambricon-X: An accelerator for sparse neural networks," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct 2016, pp. 1–12.

[6] V. Gokhale, A. Zaidy, A. X. M. Chang, and E. Culurciello, "Snowflake: An efficient hardware accelerator for convolutional neural networks," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[7] A. X. M. Chang and E. Culurciello, "Hardware accelerators for recurrent neural networks on FPGA," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[8] A. Aimar, H. Mostafa, E. Calabrese, A. Rios-Navarro, R. Tapiador-Morales, I. Lungu, M. B. Milde, F. Corradi, A. Linares-Barranco, S. Liu, and T. Delbruck, "Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2018.

[9] C. Gao, D. Neil, E. Ceolini, S.-C. Liu, and T. Delbruck, "DeltaRNN: A power-efficient recurrent neural network accelerator," in *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '18. NY, USA: ACM, 2018, pp. 21–30.

[10] N. Zimmerman, A. A. Presto, S. P. N. Kumar, J. Gu, A. Hauryliuk, E. S. Robinson, A. L. Robinson, and R. Subramanian, "A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring," *Atmospheric Measurement Techniques*, vol. 11, pp. 291–313, 2018.

[11] D. B. Topalović, M. D. Davidović, M. Jovanović, A. Bartonova, Z. Ristovski, and M. Jovašević-Stojanović, "In search of an optimal in-field calibration method of low-cost gas sensors for ambient air pollutants: Comparison of linear, multilinear and artificial neural network approaches," *Atmospheric Environment*, 6 2019.

[12] S. D. Vito, M. Piga, L. Martinotto, and G. D. Francia, "CO, NO2 and NOx urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization," *Sensors and Actuators B: Chemical*, vol. 143, pp. 182–191, 2009.

[13] S. Marco and A. Gutierrez-Galvez, "Signal and data processing for machine olfaction and chemical sensing: A review," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3189–3214, 1 2012.

[14] A. Romain and J. Nicolas, "Long term stability of metal oxide-based gas sensors for e-nose environmental applications: An overview," *Sensors and Actuators B: Chemical*, vol. 146, no. 2, pp. 502 – 506, 2010.

[15] J. Burgués, J. M. Jiménez-Soto, and S. Marco, "Estimation of the limit of detection in semiconductor gas sensors through linearized calibration models," *Analytica Chimica Acta*, vol. 1013, pp. 13 – 25, 2 2018.

[16] S. D. Vito, E. Esposito, M. Salvato, O. Popoola, F. Formisano, R. Jones, and G. D. Francia, "Calibrating chemical multisensory devices for real world applications: An in-depth comparison of quantitative machine learning approaches," *Sensors and Actuators B: Chemical*, vol. 255, pp. 1191–1210, 2018.

[17] R. Laref, E. Losson, A. Sava, K. Adjallah, and M. Siadat, "A comparison between SVM and PLS for e-nose based gas concentration monitoring," *2018 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1335–1339, 2018.

[18] Ł. Lentka, J. M. Smulko, R. Ionescu, C. G. Granqvist, and L. B. Kish, "Determination of gas mixture components using fluctuation enhanced sensing and the LS-SVM regression algorithm," *Metrology and Measurement Systems*, vol. 22, pp. 341–350, 2015.

[19] S. D. Vito, E. Massera, M. Piga, L. Martinotto, and G. D. Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2 2008.

[20] J. G. Casey, A. Collier-Oxandale, and M. Hannigan, "Performance of artificial neural networks and linear models to quantify 4 trace gas species in an oil and gas production region with low-cost sensors," *Sensors and Actuators B: Chemical*, no. Meas. Tech. 7 10 2014, 12 2018.

[21] J. Burgués and S. Marco, "Multivariate estimation of the limit of detection by orthogonal partial least squares in temperature-modulated mox sensors," *Analytica Chimica Acta*, vol. 1019, pp. 49 – 64, 2018.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[23] Z. Wen, J. Shi, Q. Li, B. He, and J. Chen, "ThunderSVM: A fast SVM library on GPUs and CPUs," *Journal of Machine Learning Research*, vol. 19, pp. 1–5, 2018.

[24] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[25] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[26] R. Bhardwaj, S. Sinha, N. Sahu, S. Majumder, P. Narang, and R. Mukhiya, "Modeling and simulation of temperature drift for ISFET-based pH sensor and its compensation through machine learning techniques," *International Journal of Circuit Theory and Applications*, vol. 47, no. 6, pp. 954–970, 2019.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[29] *FIS GAS SENSOR SB-500-12 for CARBON MONOXIDE DETECTION*, Nissha FIS, Inc., 10 2018.

[30] *TGS 3870-A04 - for the detection of Carbon Monoxide*, FIGARO USA, Inc., 2017.

[31] C. Gao, A. Rios-Navarro, X. Chen, T. Delbruck, and S. Liu, "Edge-DRNN: Enabling low-latency recurrent neural network edge inference," in *2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS2020)*, Genova, Italy, 2020.